Technical Design Report

Eliot Berry, Jacob Bongaarts, Zoe Dearing, Alex Frederickson, Pavel Golenkov, Hayden Hirsch, Will Kheriaty, Jos Kucel, Anne Le, Daniel Marchione, John McGhee, Eli Ng, Eloise Pelham, Kyan Schmidt, Duncan Sloan, Daniel Staats

Abstract

The Gibson Ek Circuit Trees are excited to present our AUV, Sir Swims-a-Lot, for the second year. Our strategy going into this year was to use the full functionality of the robot. This was also the year where thought out organizational design and established a leadership structure, a vital step in future-proofing the club. Before we added any parts to our robot, we wanted to make sure we were maximizing our current robot, so while the robot is nearly identical physically, its software and the team's understanding have significantly improved.

Competition strategy

We examined each task and determined which ones we could complete with limited mechanical improvements, focusing on software changes and the use of previously unutilized sensors.

1. ***Collecting Data***

We completed this task last year with no needed hardware, and we will reuse most of last year's code to complete this.

2. ***Navigate the Channel***

To complete this task, we will implement a few key functionalities. First, we integrated the sub's camera and identified the slalom markers. Next, we will need to navigate the sub between these markers. This will require spatial awareness and accurate maneuvering.

3. ***Ocean Cleanup***

The Ocean Cleanup task has one sub-task that we targeted: surfacing inside the octagon. This will require similar functionality to Navigate the Channel, but would need the new function of identifying an octagon shape.

4. ***Return Home***

This challenge does not require any additional functionality.

Design strategy

We are using an off-the-shelf BlueRov2 kit, the same robot we used last year. We chose this because we wanted the best chance of succeeding in the competition, and we didn't have the time to build a robot from scratch last year. This year, we are focused on optimizing last year's sub for competition. We have not made any major mechanical changes this

year, instead diving into the software side and building upon lessons learned last year.

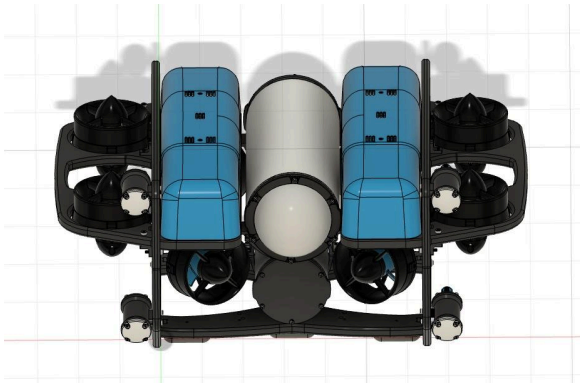

Fig 1. AUV Model

Fig 2. AUV Model

1. *Leak sensors*

We integrated the sub's onboard leak sensors into our code to prevent damage. The leak sensors are located on the hull interior and detect moisture.

2. *Kill Switch*

The sub did not come with a kill switch, which was one of our first improvements. We landed on a Blue Robotics MOSFET PCB, a kill switch that intercepts power from the battery and can handle the large currents we are working with.

3. *Wet Release Tether*

The Blue Robotics ROV is intended to be used while tethered and is not configured for autonomous use. We usually load code onto the robot while tethered, but often need to run untethered tests. With its off-the-shelf configuration, we need to open the robot to remove the tether, a time-consuming process that is especially difficult during a pool test. We spliced the robot's tether and implemented a wet release. This allows us to detach and reattach the tether without needing to open the hull.

Software Systems

After doing research on our options, we opted to use Blue OS, the native software of our sub. It has a robust API that is compatible with multiple coding languages. This flexibility and ease made it a perfect fit for the project. We landed on Python primarily because of its accessibility to newer coders and functionality.

1. *Code Base Improvements*

The first major improvement we made to our code was cleaning it up and setting best practices. Much of our existing code base was written last-minute at the competition the previous year. It lacked the documentation and standard conventions we would need to keep a project of this scale manageable. Cleaning and documenting this code was essential to further development.

We had a large influx of new members interested in working on the robot's software, some being first-time coders. We established rules for the GitHub repository and wrote up a best practices document to minimize future issues.

2. Camera Utilization & Color Sensing

In order to complete the Navigate the Channel task and surfacing inside the octagon in Ocean Clean Up, we would need to interface with the sub's onboard camera.

For the slalom task, we identified where the channel markers were with color detection through Python's CV2 module. Once identified, we plotted a trajectory that threaded between them.

3. Compass Utilization

Getting accurate directional readings is vital to keeping the robot on course, especially in the slalom task. Our sub had an onboard magnetometer which we had not utilized last year. We processed its outputs, converting them into a usable format, and integrated them into the navigation process.

We held all of our in-water testing at Julius Boehms Pool, a pool next door to our school. The only time slot we were able to reserve was 5:30 AM- 6:30 AM on Fridays in their smaller pool. Since we had a limited amount of time to test, we wanted to make the most of it. We did two important things to optimize our testing time.

The first major improvement we made to streamline testing was adding a wet release tether to our sub. This let us run untethered tests without the hassle of opening the hull to disconnect the tether, saving valuable time.

The second improvement was building the 'charcuterie board', a wooden board with spare parts and identical wiring to the sub. This let us test electronic components quickly and have replacements prepared in case a part fails. Additionally, it gave us the ability to experiment with code while the robot was disassembled, which was fairly often.

We established testing procedures and defined specific roles and responsibilities to ensure our pool tests went smoothly and to keep our sub safe.

Testing Strategy

Component	**Vendor**	**Custom/Purchased**	**Cost**	**Year**
BlueROV2	Blue Robotics	Purchased	$4250	2024

BlueROV2 Heavy Configuration Retrofit Kit	Blue Robotics	Purchased	$790	2024
Fathom ROV Tether (ROV-ready)	Blue Robotics	Purchased	$400	2024
Lumen Subsea Light for ROV/AUV	Blue Robotics	Purchased	$160	2024
Xbox Wireless Controller	Blue Robotics	Purchased	$65	2024
BlueROV2 Spares Kit	Blue Robotics	Purchased	$350	2024
Subsea Buoyancy Foam: R-3312 (16 in x 8 in x 1 in) x2	Blue Robotics	Purchased	$140	2024
Subsea Buoyancy Foam: R-3312 (24 in x 8 in x 2.5 in) x2	Blue Robotics	Purchased	$320	2024
Newton Subsea Gripper	Blue Robotics	Purchased	$640	2024
H6 PRO Lithium Battery Charger	Blue Robotics	Purchased	$185	2024
Lithium-ion Battery (14.8v, 18Ah)	Blue Robotics	Purchased	$380	2024
BlueROV2 MOSFET Battery Switch	FieldWerx	Purchased	$125	2024

[1]A. Amer, O. Álvarez-Tuñón, H. Ugurlu, J. Le Fevre Sejersen, Y. Brodskiy, and E. Kayacan, "UNav-Sim: A Visually Realistic Underwater Robotics Simulator and Synthetic Data-generation Framework." https://arxiv.org/pdf/2310.11927

[2]"Docs | Home," *blueos.cloud*. https://blueos.cloud/docs/

[3]"ROS 2 Documentation — ROS 2 Documentation: Rolling documentation," *docs.ros.org*. https://docs.ros.org/en/rolling/index.html

[4]"BlueROV2 Assembly," *Blue Robotics*. https://bluerobotics.com/learn/bluerov2-assembly/